

Amendments to the Claims:

Please cancel claims 1, 3, 5, 6, 10, 13, 16, 17, 20, 24, 25, 26 and 28-38. Please amend claims 2, 4, 7, 8, 11, 14, 18, 19, 21 and 27 as noted below. Please add newly presented claims 39-41.

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Canceled)
2. (Currently Amended) The method of claim 4 39, further comprising the steps of:  
initializing callback function pointers for use by the application as callback functions into the ODK subsystem; and  
initializing the extension after the extension is loaded.
3. (Canceled)
4. (Currently Amended) The method of claim 4 39, further comprising the step of checking whether a stop to run transition has occurred in the PLC and if so, sending a specific formatted activate application message from the PLC to the ODK subsystem.
5. (Canceled)
6. (Canceled)
7. (Currently Amended) The method of claim 6 39, further comprising the step of executing the generic execution request by the application.

8. (Currently Amended) The method of claim ~~6~~ 39, further comprising the steps of:  
sending a generic response from the application to the ODK subsystem;  
converting the generic response to a specific formatted response; and  
sending the specific formatted response from the ODK subsystem to the PLC.
9. (Original) The method of claim 8, further comprising the step of returning at least one of data and a control block from the application to the ODK subsystem, and from the ODK subsystem to the PLC.
10. (Canceled)
11. (Currently Amended) The method of claim ~~40~~ 39, wherein in the requesting information step the application uses a call back pointer to generically request information and the executing step executes the function in the ODK subsystem corresponding to the callback pointer.
12. (Original) The method of claim 11, in the executing a function step, the function is provided by a dynamic link library.
13. (Canceled)
14. (Currently Amended) The method of claim ~~40~~ 39, wherein when the checking determines that there are no requests for information waiting, and further comprising:  
waiting until the PLC transitions from a run state to a stop state;  
sending a deactivate request from the PLC to the ODK subsystem; and  
calling a deactivate function in the application.
15. (Original) The method of claim 14, wherein when a memory clear or PLC shutdown occurs, the step of calling a release function in the application and unloading the extension occurs.

16. (Canceled)

18. (Currently Amended) The system of claim ~~47~~ 40, wherein the ODK subsystem further comprises: further comprising:

a system block loader adapted to load system blocks, the system blocks including at least one of a system function block, a system function, a system data block; and

~~an the ODK SB Add-on dynamic link library (DLL) for implementing~~ DLL implements a common object module (COM) interface for the virtual CPU and the system block loader.

19. (Currently Amended) The system of claim ~~47~~ 40, wherein the one or more extensions are dynamic link libraries.

20. (Canceled)

21. (Currently Amended) The system of claim ~~20~~ 40, further comprising the steps of:

a means for initializing callback function pointers for use by the application as callback functions into the ODK subsystem; and

a means for initializing the extension after the extension is loaded.

22. (Currently Amended) The system of claim ~~20~~ 40, further comprising a means for checking whether a stop to run transition has occurred in the PLC and if so, a means for sending a specific activate application message from the PLC to the ODK subsystem.

23. (Original) The system of claim 22, further comprising a means for calling an activate function in the application by the ODK subsystem thereby permitting scan cycle execution.

24. (Canceled)

25. (Canceled)

26. (Canceled)

27. (Currently Amended) The system of claim ~~26~~ 40, further including a means for returning at least one of data and a control block from the application to the ODK subsystem, and from the ODK subsystem to the PLC.

28. (Canceled)

29. (Canceled)

30. (Canceled)

31. (Canceled)

32. (Canceled)

33. (Canceled)

34. (Canceled)

35. (Canceled)

36. (Canceled)

37. (Canceled)

38. (Canceled)

Please add the following new independent claims 39-41 as follows:

39. (Newly Presented) In a control automation system having a computer having a processor, computer memory accessible by the processor, and an expansion bus capable of coupling to and communicating with peripheral devices; at least one programmable logic controller (PLC) in communication with the computer, the PLC being any one or more of a hard PLC, slot PLC or soft PLC configuration device, the PLC being capable of performing application tasks; a soft PLC having a virtual CPU run by the computer processor, and adapted to execute a PLC application program in a real-time operating environment and adapted to execute one or more extensions that provide access into a scan cycle of the soft PLC and provide replaceable functionality to an operation of the PLC; and an open development kit (ODK) subsystem in communication with the computer and the PLC, and having an Application Wizard that is capable of accessing and causing execution of at least one PLC application program, the Application Wizard being adapted to provide a first operational mode useable in a WinAC RTX deterministic environment to generate a Real Time Dynamic Link Library project with fixed scan cycles, including an ODK RTX Proxy DLL, and a second operational mode adapted to generate a Dynamic Link Library project that is useable in a non-deterministic environment with non-fixed scan cycles, including an ODK SB Add-on DLL, wherein the ODK RTX Proxy DLL and the ODK SB Add-on DLL are in communication through shared memory; a method for operating the ODK subsystem comprising:

checking whether there are requests for application execution in at least one PLC that is in communication with the ODK, and if so, causing the respective PLC to send the application execution request message that is specifically formatted for the respective PLC configuration device to the ODK subsystem, the ODK subsystem converting the specific formatted application execution request message to a generic formatted application execution request message, and sending the generic formatted application execution request message to the Application Wizard; and

checking whether any requests for information are waiting in the executed application, and, if so, requesting information from the at least one PLC on behalf of the executed application, executing a function in the ODK subsystem that is specified by the application,

causing the respective PLC to perform the task that is associated with the executed function in the ODK subsystem, and returning a specific formatted response from the respective PLC to the ODK subsystem; the ODK subsystem in turn converting the specific formatted response to a generic response and returning the generic response from the ODK subsystem to the executed application.

40. (Newly Presented) A control automation system comprising:  
a computer having a processor, computer memory accessible by the processor, and an expansion bus capable of coupling to and communicating with peripheral devices;  
at least one programmable logic controller (PLC) in communication with the computer, the PLC being any one or more of a hard PLC, slot PLC or soft PLC configuration device, the PLC being capable of performing application tasks;  
a soft PLC having a virtual CPU run by the computer processor, and adapted to execute a PLC application program in a real-time operating environment and adapted to execute one or more extensions that provide access into a scan cycle of the soft PLC and provide replaceable functionality to an operation of the PLC; and  
an open development kit (ODK) subsystem in communication with the computer and the PLC, and having an Application Wizard that is capable of accessing and causing execution of at least one PLC application program, the Application Wizard being adapted to provide a first operational mode useable in a WinAC RTX deterministic environment to generate a Real Time Dynamic Link Library project with fixed scan cycles, including an ODK RTX Proxy DLL, and a second operational mode adapted to generate a Dynamic Link Library project that is useable in a non-deterministic environment with non-fixed scan cycles, including an ODK SB Add-on DLL, wherein the ODK RTX Proxy DLL and the ODK SB Add-on DLL are in communication through shared memory, wherein:

the ODK subsystem is capable of checking whether there are requests for application execution in any one of a hard PLC, a slot PLC or a soft PLC, and if so, causing the respective PLC to send the application execution request message that is specifically formatted for the respective PLC configuration device to the ODK subsystem, the ODK subsystem converting the specific formatted application execution request message to a

generic formatted application execution request message, and sending the generic formatted application execution request message to the Application Wizard of the ODK;  
and

the ODK subsystem is also capable of checking whether any requests for information are waiting in the executed application, and, if so, requesting information from the respective PLC on behalf of the executed application, executing a function in the ODK subsystem that is specified by the application, causing the respective PLC to perform the task that is associated with the executed function in the ODK subsystem, and returning a specific formatted response from the respective PLC to the ODK subsystem; the ODK subsystem in turn converting the specific formatted response to a generic response and returning the generic response from the ODK subsystem to the executed application.

41. (Newly Presented) A computer storage medium storing a computer program adapted for use in a control automation system computer having a processor and computer memory accessible by the processor, and an expansion bus capable of coupling to and communicating with peripheral devices; wherein there is at least one programmable logic controller (PLC) in communication with the computer, the PLC being any one or more of a hard PLC, slot PLC or soft PLC configuration device, the PLC being capable of performing application tasks; the computer program including a soft PLC having a virtual CPU capable of being run by the computer processor, and adapted to execute a PLC application program in a real-time operating environment and adapted to execute one or more extensions that provide access into a scan cycle of the soft PLC and provide replaceable functionality to an operation of the PLC; and the computer program further having an open development kit (ODK) capable of being run by the computer processor as an ODK subsystem and enabling communication between the computer and the PLC, the ODK computer program also having an Application Wizard that when run by the processor is capable of accessing and causing execution of at least one PLC application program, the Application Wizard being adapted to provide a first operational mode useable in a WinAC RTX deterministic environment to generate a Real Time Dynamic Link Library project with fixed scan cycles, including an ODK RTX Proxy DLL, and a second operational mode adapted to generate a Dynamic Link Library project that is useable in a non-deterministic

environment with non-fixed scan cycles, including an ODK SB Add-on DLL, wherein the ODK RTX Proxy DLL and the ODK SB Add-on DLL are in communication through shared memory; the software code when run by the processor enables the ODK subsystem to perform the steps of:

checking whether there are requests for application execution in at least one PLC that is in communication with the ODK subsystem, and if so, causing the respective PLC to send the application execution request message that is specifically formatted for the respective PLC configuration device to the ODK subsystem, the ODK subsystem converting the specific formatted application execution request message to a generic formatted application execution request message, and sending the generic formatted application execution request message to the Application Wizard; and

checking whether any requests for information are waiting in the executed application, and, if so, requesting information from the at least one PLC on behalf of the executed application, executing a function in the ODK subsystem that is specified by the application, causing the respective PLC to perform the task that is associated with the executed function in the ODK subsystem, and returning a specific formatted response from the respective PLC to the ODK subsystem; the ODK subsystem in turn converting the specific formatted response to a generic response and returning the generic response from the ODK subsystem to the executed application.